

| | |
|-------------------|-----------------|
| PH US 000184WD | MAT. DOSSIER |
|-------------------|-----------------|

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
14 December 2000 (14.12.2000)

PCT

(10) International Publication Number
WO 00/76131 A1

(51) International Patent Classification⁷: H04L 12/28

(21) International Application Number: PCT/EP00/05026

(22) International Filing Date: 31 May 2000 (31.05.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
99401329.0 2 June 1999 (02.06.1999) EP

(71) Applicant (for all designated States except US): THOMSON MULTIMEDIA [FR/FR]; 46, quai Alphonse Le Gallo, F-92100 Boulogne-Billancourt (FR).

(72) Inventors; and

(75) Inventors/Applicants (for US only): BICHOT, Guillaume [FR/FR]; Thomson Multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR). BURKLIN, Helmut [DE/FR]; Thomson Multimedia, 46, quai

Alphonse Le Gallo, F-92648 Boulogne Cedex (FR). RAMASWAMY, Kumar [IN/FR]; Thomson Multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR). STAHL, Thomas [US/FR]; Thomson Multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR).

(74) Agent: KOHRS, Martin; Thomson Multimedia, 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR).

(81) Designated States (national): AU, BG, BR, CA, CN, CZ, HU, ID, IN, JP, KR, MX, NZ, PL, SG, SI, SK, TR, US, VN, YU, ZA.

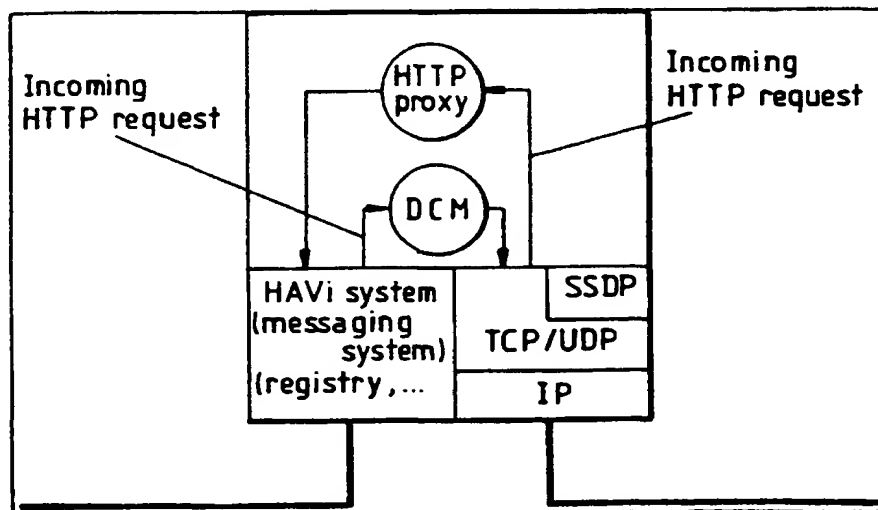
(84) Designated States (regional): Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published:

- With international search report.
- Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.

[Continued on next page]

(54) Title: METHODS FOR BRIDGING A HAVi SUB-NETWORK AND A UPnP SUB-NETWORK AND DEVICE FOR IMPLEMENTING SAID METHODS



(57) Abstract: The invention concerns methods for bridging a HAVi sub-network and a UPnP sub-network. For making UPnP devices available to the HAVi sub-network, the following steps are implemented: discovering UPnP devices and/or services on the UPnP sub-network; declaration, by a sub-network bridging device, of a discovered UPnP device as a HAVi Device Control Module and of a discovered UPnP service as a HAVi Functional Component Module on the HAVi sub-network. Formaking HAVi devices available to the UPnP sub-network, the following steps are implemented: discovering HAVi software elements of the HAVi sub-network corresponding to a selection criterion; representing, in a sub-network bridging device, each of said discovered elements by a UPnP proxy service identified by a port number attributed by said sub-network bridging device; and announcing each of said proxy services on the UPnP sub-network. The invention also concerns a device for implementing the above steps.

WO 00/76131 A1



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Methods for bridging a HAVi sub-network and a UPnP sub-network and device for implementing said methods

5 The invention concerns methods to bridge HAVi and UPnP sub-networks, a first method addressing the issue of making UPnP objects available to HAVi objects, a second method addressing the issue of making HAVi objects available to UPnP objects. The invention also concerns a bridge device.

 The invention applies among others to home networks.

10 Several sub-networks based on different physical media (wired and wireless) and applications are expected to coexist in a digital home network. Common examples of wired physical media include the coaxial cable, twisted pair wiring, power line and optical fibers. A digital home network also needs to contend with the technological
15 developments within the computer, consumer electronics, telephony and home automation industries.

 There have been two recent initiatives within the industry to address different needs:

20

1. Home Audio-Video interactivity (HAVi)
2. Universal Plug and Play (UPnP)

25 There is a need for harmonization of the two system approaches in order to ensure coexistence and interoperability of devices within these domains. The bridging of the two technological approaches. is thus desirable.

30 The first initiative, Home Audio-Video Interoperability (HAVi), started within the consumer industry is an attempt to accomplish high-speed interconnectivity over an IEEE 1394 serial bus network for transacting audio/visual data. This initiative was specifically intended to address the needs of the consumer electronics devices to enable interactivity (with user involvement in a user-device interaction paradigm) and interoperability (with no user involvement in a device-device interaction paradigm). Further, within HAVi, there is a strong emphasis on enabling streaming applications in
35 addition to control applications. An example of a streaming application would be an application transferring a video stream from a recording device to a decoder or display, while an example of a control application would be an application for programming the behavior of devices. This implies a support for both isochronous and asynchronous transactions.

40

The other key features of HAVi include:

1. Hot Plug and Play
2. Hardware and Operating System Platform neutrality
- 5 3. Support for legacy devices (i.e. devices with no HAVi functionality) and a gradual evolution to support new technologies

10 The second initiative is Universal Plug and Play (UPnP). While HAVi is intended primarily for a high speed IEEE 1394 network for Audio/Video transactions, UPnP is an initiative that is physical layer agnostic and expects to work on a TCP/IP network. The general notions and paradigms are based on the Internet protocols with additions to support the notions of plug and play.

15 The following non-exhaustive list of documents describe the above architectures in their current state of development, at the priority date of the present application:

For HAVi:

20 The HAVi 1.0 beta+ specification, dated October 23, 1998

The UPnP technology comprises a set of protocols based on TCP/IP:

- 'Automatically choosing an IP Address in an Ad-Hoc IPv4 Network' (<draft-ietf-dhc-ipv4-autoconfig-04.txt>)
- 'Simple Service Discovery Protocol 1.0' (<<http://search.ie.org/internet-drafts/draft-cai-ssdp-v1-01.txt>>)
- 25 - 'Multicast Discovery of DNS (Domain Name Server) Services' (<draft-manning-multicast-dns-01.txt>).
- 'Extended Markup Language' - XML (1.0 W3C recommendation)

30 More information concerning these two architectures is available on the corresponding websites, ie. www.HAVi.org and www.UPnP.org.

35 At present, there is no interoperability to ensure that a uniform control paradigm, i.e. model, is available so that devices in both the HAVi sub-network and the UPnP sub-network are able to interact with devices and perform control functions over their respective sub-network boundary.

The object of the invention is a method for bridging a HAVi sub-network and a UPnP sub-network comprising the steps of:

- discovering UPnP devices and/or services on the UPnP sub-network;

5 - declaration, by a sub-network bridging device, of a discovered UPnP device as a HAVi Device Control Module and of a discovered UPnP service as a HAVi Functional Component Module on the HAVi sub-network.

UPnP devices and/or services normally all have HTTP capability.

10 According to a variant, only certain UPnP devices or services are declared on the as HAVi DCMs or FCMs, based on selection criteria.

According to a preferred embodiment, the discovery step is carried out using Simple Service Discovery Protocol (SSDP) functions.

15 Another object of the invention is a method for bridging a HAVi sub-network and a UPnP sub-network comprising the steps of:

- discovering HAVi software elements of the HAVi sub-network corresponding to a selection criterion;

20 - representing, in a sub-network bridging device, each of said discovered elements by a UPnP proxy service identified by a port number attributed by said sub-network bridging device; and

- announcing each of said proxy services on the UPnP sub-network.

25 According to a preferred embodiment, the discovery step comprises the step of requesting, by said sub-network bridging device, a list of software elements from its HAVi Registry.

30 According to a preferred embodiment of the inventions above, the selection criterion is HTTP capability.

HTML capability may be another criterion.

Another object of the invention is a device for bridging a UPnP sub-network and a HAVi sub-network implementing the above method steps.

35 Other characteristics and advantages of the invention will become apparent through the description of a non-limiting, preferred embodiment, described with the help of the attached drawings among which:

- figure 1, already described, represents a HAVi network and a UPnP network linked through a bridge device ;

- figure 2, already described, represents a single network comprising both HAVi and UPnP compliant devices ;

5 - figure 3 represents the networks of figure 1 in which each network comprises an HTML browser capable device ;

- figure 4 is a schematic diagram of protocol stacks in the bridge device,

- figure 5 is a flowchart of the process carried out to represent a UPnP device in the network,

10 - figure 6 is a flowchart of the process carried out to represent a HAVi device in the network.

In addition to the documents cited in the introduction of the present application, one should also refer to the Hypertext Transfer Protocol ('HTTP') 1.1 for
15 further background information.

Figure 1 represents an example of a home network composed of a HAVi based home sub-network and a UPnP based home sub-network that are bridged together. Nodes A and D are displays where the user can view the network topology
20 and can control, through an appropriate user interface, any node on either network. This implies that from node A, the user should for example be able to detect the connection of node E to the UPnP network and should be able to control it. In a similar manner, a user of node D on the UPnP sub-network should be able to detect the appearance of a new HAVi device within the HAVi sub-network and should be able to
25 control it.

In Figure 1, the two sub-networks are built over two different media. However, the problem to be solved is the same in the situation where both network architectures are built over the same media as shown in Figure 2. The nodes A, B and
30 C are HAVi aware and the nodes C, D and E are UPnP aware. In both configurations, there is a node (node C in both examples) which implements the bridge function in order to enable the control of any device across the entire network.

Within the context of the UPnP network, it is necessary to detect the entry of
35 a new device into the network, announce its capabilities in a well defined manner and allow the commencement of interaction with other devices within the network. The

SSDP protocol and XML operating over the TCP/IP network are used to accomplish this functionality.

For user control of devices, HTML may be used instead of XML.

5 As described earlier, HAVi is a complete system solution to the interoperability of devices with a IEEE 1394 interface. HAVi addresses, among others, the following issues:

1. Registration of devices and of functions thereof
- 10 2. Communication media management (the media being an IEEE 1394 serial bus in this case)
3. Representing devices and functions within devices using device control modules ('DCMs') and function control modules ('FCMs'), respectively:
- 15 - the list of services provided by a DCM includes connection management, status queries for the device and its plugs
- there is a comprehensive list of FCMs representing the most common consumer electronic functional elements
- 20 4. Event management (using an event manager)
5. Stream management (using a stream manager)
6. Resource management
7. Support for legacy devices
8. Data Driven Interaction (DDI) mechanism

25

Additions to HAVi compared to UPnP include specific provisions for stream management, communication media management and resource management. These additions are quite important in the context of an A/V network based on the IEEE 1394
30 bus, which imposes severe real time constraints.

One of the issues regarding the interaction between the HAVi and UPnP sub-networks is the need for a common user control protocol. The Data Driven Interaction (DDI) protocol, which is standard in HAVi, is not standard within the UPnP
35 network. According to the DDI protocol, a controller software element obtains data relative to a user interface from a target software element. The controller uses this data to build a user interface. HTML on the other hand is standard in an Internet Protocol (IP) based computer network (including UPnP), but is not included in HAVi. The following functionalities are thus required:

- a user control protocol which can be used across the whole network. The embodiment uses the HyperText Mark-up Language (HTML) to implement the user control protocol. Another protocol with similar functionality may of course be used.

- a bridge function which will allow :

- 5 (a) connection of the two different discovery protocols
(b) carrying the user control protocol data over the entire network, HTML data in the present case.

10 The Hypertext Transfer Protocol (or HTTP) is a request/response protocol. The embodiment uses this protocol to accomplish the bridging function mentioned earlier.

The request/response mechanism dictates a client/server model for devices using HTTP. Two objects are involved in HTTP : the client, which sends a command, and an origin server that receives the command and sends back the response.

15 The HTTP protocol has a list of well defined methods (or commands) which include the following:

- 20 1. Option
2. Get
3. Head
4. Post
5. Put
6. Delete
7. Trace

25

The most commonly used command is *GET < URL>* where the Uniform Resource Locator (URL) points towards the object to be obtained. This reference is composed of two parts : the first points towards the server equipment and the second points towards the object associated with the command. This target object can be an existing object such as a HTML script or a bit-map or other objects. However the object reference can point towards something that has a meaning for the server but does not represent any real object. This is used, for instance, in an HTML script to signal to the server that the user just selected an icon. After the user selects the icon, the HTML script associates this icon with a URL which will be sent to the WEB server (through the GET command). A URL reference can thus include some parameters representing a command from upper layer protocols.

30
35

Basically, a user (which can be an application) requires the ability to detect the network changes (new or removed devices and/or services). He also requires the ability to control devices through a user interface (a well-known language or protocol).

5 The user control interface model of both networks is first addressed. Then, a bridge between other control protocols to operate UPnP and HAVi services over the entire network is defined.

10 Each network technology specifies a way to dynamically discover the appearance or the disappearance of services and devices. The first task of the bridge function is to connect both discovery methods, through

- a method to represent and to reach a UPnP device/service for a HAVi application within the bridge device;
- 15 - a method to represent and to reach a HAVi device/service for a UPnP application within the bridge device.

20 The UPnP protocol for this discovery function is the Simple Discovery Protocol (called SSDP) mentioned in an earlier section. The HAVi protocol for the corresponding function is the Registry.

25 Regarding the bridging of the user control interface we need to map the HAVi and the UPnP worlds. HAVi specifies two protocols which are different from the user control protocol HTML.

30 The present embodiment proposes to define:

- a user control protocol which can be used across the whole network : HTML over HTTP is chosen, since it is already available in one of the sub-networks;
- a method to implement the chosen user control protocol (HTTP) protocol within the HAVi paradigm.

35 Regarding the bridging of other services (UPnP or HAVi), the embodiment includes:

- a method for a HAVi applications to operate the UPnP service/device;
- a method for a UPnP application to operate the HAVi service/device.

In Figure 3, the bridge function is included in the device C. Any device, irrespective of its functionality, could host the bridge function. The host device

comprises the HAVi middleware and the UPnP protocol stack. For clarity purposes, the C device according to the present embodiment does not provide any functionality other than the bridge as described below.

5 The next section describes the steps to take into account the following scenarios:

10 - Scenario 1: the network topology is the same as shown in the Figure 3 except that the E device is not present. After power on, a user application of A (e.g. an HTML browser) is able to give back to the user the network list of the HTML controllable devices. This includes all UPnP sub-network devices and the HTML capable devices of the HAVi sub-network. The E device is plugged on to the UPnP network. The User application detects this new device (through the ANNOUNCE method of SSDP). The user should then be able to control the E device using HTML.

15 - Scenario 2: the network topology is the same as shown in the Figure 3 except that the B device is not present. After power on, the User application of D (the HTML browser), is able to give back to the user the network list of the HTML controllable devices (discovery by SSDP queries). The B device is plugged on to the HAVi network. The user application detects this new device (announcement). Then the user will be able to control the B device using HTML.

25 HTML data is transported using the HTTP protocol. However, HTTP is a means to transport any type of hyper text based content. Just as through HTTP we are able to obtain any User Interface object, such as an icon or a bitmap, we are able to transport XML content as well. Today, the most popular Markup language is HTML. Consequently many product tools already exist. This is the reason that according to the present embodiment, HTML is implemented within the HAVi network. XML is an emerging standard, and could be used instead of HTML without any modification to the embodiment since XML content can also be transported over HTTP . According to a variant embodiment, XML and HTML are used jointly. Other protocols may be used as well.

35 The following section describes one implementation of HTTP (HTML) in the HAVi sub-network. This protocol is a simple command/response protocol between a controller and a target (called HTTP server). In HAVi, each device is represented by a software component called a DCM (Device Control Module). This DCM contains a certain number of well specified entry points (represented by a set of functions) which can be used (called) by any other software element of the HAVi network. Like a C-

language function, when a software element calls a function of a DCM (whether remotely or locally, this being transparent to the caller), an associated process is started and the function returns the result of the process. To implement the HTTP paradigm, a new set of function is added within the DCM Application Programmable Interface (API) to offer the possibility to handle the HTTP protocol between two HAVi software elements - for example between an application (a browser) and a DCM (the HTTP server). HAVi uses the IDL (Interface Definition Language) to specify a function. Due to the nature of the HTTP protocol, it is possible that HTTP requests or responses contain a very large payload. The transport layer of HAVi specifies a limit on the message size that can be exchanged between two HAVi software elements. However HAVi specifies a way to handle such large messages by a recommended design of the API of the element potentially involved in such communications (see APIs for Bulk transfer).

The following code illustrates the proposed API extension for the DCMs which would implement an HTTP server.

```
enum FileLoc {START, MIDDLE, END};
```

This parameter permits to indicate whether the message from a producer to a consumer is the first message, an intermediate message, or the last message. If the stream to be sent fits into one message, the END value will be used.

(a) The following function allows a software element client (or HTTP client) to open a HTTP connection with a DCM.

```
Status DCM::HttpOpen (
    in short clientBufferSize,
    in OperationCode opCode,
    out long cid,
    out short ServerBufferSize)
```

The parameters have the following meaning:

"clientBufferSize" : indicates the maximum size (in bytes) of a message accepted by the requester. The DCM will take that parameter into account during the sending of the HTTP response to the client.

"opCode" : this is the operation code the DCM will use to send the HTTP response back to the client. The client function identified by this operation code must comply with the prototype named <client>::HttpResponse, given below.

5 "cid" : the identifier of this HTTP connection with the DCM. It allows several connections from the same software element client and also permits to match a response with a request.

10 "ServerBufferSize" : indicates the maximum size (in bytes) of a message accepted by the DCM. The HTTP client will take this parameter into account during the sending of the HTTP requests.

The error codes for this function are the following:

15 "DCM::ENUM_CONN": maximum number of opened connections is reached for this DCM

"DCM::EALLOC": resource allocation error

(b) A second function is used to allow a software element client to close a HTTP connection:

20 Status DCM::HttpClose (in long cid)

The parameter "cid" is the identifier of this HTTP connection with the DCM. This function is used to close a connection with a Web proxy FCM. The error code is the following:

"DCM::ECID": The cid is unknown.

(c) A third function allows a software element client (or HTTP client) to send a request to a DCM acting as a HTTP server according to HTTP.

30 Status DCM::HttpRequest (
in long cid,
in FileLoc where,
in sequence<octet> data)

35 Parameters

"cid": the identifier of the connection between the HTTP client and the DCM (acting as a HTTP server) obtains by the HTTP client from a DCM::HttpOpen call.

"where": informs the DCM that this message is the first, the last or an intermediate message of the request.

5 "data": contains a part or the entire request according to the HTTP protocol.

Error codes:

10 "DCM::ESIZE": the data exceeds the size of the buffer in the receiver. The receiver has not received or processed the data. It is left to the implementation how the sender reacts to this status error.

"DCM::EFAILED": the receiver has aborted the transfer of the current sequence of data transfers. The sender shall abort the transfer of the current sequence.

"DCM::ECID": The "cid" is unknown.

15

(d) The fourth function to be implemented in the client allows the DCM (acting as HTTP server) to send back to the client a HTTP response corresponding to a previously received HTTP request from that client through the connection identified by "cid":

20

```
Status <Client>::HttpResponse(
    in long cid,
    in FileLoc where,
    in sequence<octet> data)
```

25

"cid": the identifier of the connection between the HTTP client and the DCM .

"where": informs the client that this message is the first, the last or an intermediate message of the response .

30

"webData": contains a part or the entire response according to the HTTP protocol used through the connection identified by the "cid" parameter.

35 This is the prototype of the function to be implemented in the client which allows the DCM (acting as a HTTP server) to give back to the client a HTTP response corresponding to a previously received HTTP request from that client through the connection identified by "cid".

The error codes are the followings:

"WebProxy::ESIZE": the data exceeds the size of the buffer in the receiver. The receiver has not received or processed the data. It is left to the implementation how the sender reacts to this status.

5 "WebProxy::EFAILED": the receiver has aborted the transfer of the current sequence of data transfers. The sender shall abort the transfer of the current sequence.

"WebProxy::ECID": The cid is unknown.

10 In HAVi, the designer of a target device can decide which user control protocol to implement, among the two protocols currently defined by HAVi. It is not required to provide user control capabilities as specified in HAVi. For the controller application, it is necessary to know whether a particular target is user-controllable or not. This is the goal of an attribute in the HAVi Registry. The Registry is a database where all software elements of a device are registered (including DCMs and application
15 modules). Any software element can query the database. Each time an element is added or removed, a corresponding network event is generated. In the Registry, an element is registered with a set of attributes which characterize it. One of these attributes is the GUIREQ attribute, which defines whether this element can be controlled through a user control protocol and if yes, through which protocol. The
20 possible values for the attribute are :

- NO_NEED
- DDI (the basic UI protocol in HAVi)
- HAVLET (the Java based UI protocol in HAVi)

25

The invention proposes to add a new value of this attribute:

- HTTP (the HTTP/HTML paradigm in HAVi)

30 When a user wants to control a device, its associated client application, typically an HTML browser, displays the set of network devices which are HTTP/HTML capable by querying the Registry on the corresponding GUIREQ attribute value. The user selects one of these and the client application can send the HTTP GET command towards the DCM of the selected target, according to that protocol. To send the HTTP command, the client application first establishes an HTTP connection with the target
35 DCM (calling the "DCM::HttpOpen" method) and then performs one or several calls (depending on the size of the request and the capabilities of the target DCM in term of the HAVi message size) to the DCM. The DCM then uses the "DCM: HttpRequest" to send its HTTP request. Once the target receives the command, it sends back the Home

HTML page associated with the device to the client by calling one or more times the client method called "HttpResponse". The client application (the browser) then interprets the HTML script and displays the corresponding User Interface.

5 The bridge function is implemented in a device (called the bridge device or simply the bridge) which is connected to both sub-networks as shown in Figure 3 (device C). Thus it has to contain, at least, the protocol stack as shown in the Figure 4. Since the SSDP protocol requires some multicast capabilities, the IP layer will be IGMP ('Internet Group Management Protocol') compliant.

10

The method of modelization of a UPnP device or service by the bridge seen from the HAVi sub-network will now be described.

15 The UPnP network is composed of devices that offer access to some network services. The SSDP protocol permits the discovery of the services available in the network and indirectly the device that hosts the service. The HAVi network is composed of devices that contain one or more functional components (equivalent to services in UPnP). As already mentioned, a DCM is used to represent a device, and an FCM is used to represent a functional component. In HAVi, the User control protocol is
20 managed through the DCM API.

According to the invention:

- 25 - A UPnP device is represented by a DCM in the bridge device.
- This DCM contains the extra API for HTTP.
- A UPnP service (except for the HTTP service) will be represented by a FCM in the bridge device.

30 It is mandatory to register these DCMs and FCMs through the HAVi REGISTRY service to allow any other HAVi object to discover them and, thus, to be able to operate them. The registration consists in registering the HAVi addresses of these software elements and the mandatory attributes according to the HAVi specification:

- 35 - Type of software element (either DCM or GENERIC FCM)
- HUID (unique hardware identifier : computed by the bridge device)
- Device class (LAV - meaning Legacy device)
- GUIReq (HTTP)
- DeviceManufacturer (manufacturer of the UPnP device/service)

- SoftwareElementVersion (computed by the bridge device)
- UserPreferredName (computed by the bridge device – based on the UPnP service/device name)

5 Before the bridge device is able to activate and register a DCM/FCM corresponding to a UPnP device/service, it has to detect its presence within the UPnP sub-network.

10 According to the SSDP (Simple Service Discovery Protocol) of UPnP, the bridge device acts as a SSDP client and server. Once the bridge device is connected to the home network, its SSDP client has to query the UPnP sub-network by sending the multicast "OPTIONS" message over UDP/IP to query the SSDP servers. The SSDP "OPTIONS" message will have the following format according to the HTTP specification:

15 OPTIONS _*_ HTTP/1.1 Request-ID: uuid:1313Alt-Locations: <httpu://bridge.com/bar:1111>

20 This message contains the type of services concerned by the query ("_*_" meaning : all), the version number of HTTP, a unique identifier to match the response with the query (the value shall respect a format described in the RFC 2518), and the URL the responder will use to give back its response(s) (the port number 1111 correspond to the SSDP client of the bridge).

25 All UPnP devices will send back one or several SSDP OPTIONS responses (one by service implemented within the device) containing the name of the service, the network location of the service (the URL used to reach the service), the protocol to be used to communicate with the service and a set of data to describe the device which hosts the service according to the following :

- Device manufacturer name
- Device name
- A URL to obtain an icon representing the device

30 The bridge device parses all responses and:

- For each new device detected, it installs and declares a HAVi DCM and registers it in its local Registry with its well specified attributes as described earlier.
- For each new service type, for which the bridge wants to offer the access to the HAVi network, it installs and declares a HAVi FCM and also registers it with its well

specified attributes as described earlier. Other attributes than those indicated above may also be registered.

Consequently the DCM, respectively the FCM shall maintain a set of configuration data to be able to identify and join its associated UPnP device, respectively service.

For the DCM:

- The URL to join the HTTP server of the UPnP device (if this service is implemented in the device)
- The Device manufacturer name
- The Device name
- The URL to get an icon representing the device

For the FCM:

- The URL to join the UPnP service
- The type of the UPnP service (printer for example)
- The Device manufacturer name of the UPnP host device
- The service name of the UPnP service (PrinterRoom2 for example)
- The URL to get an icon representing the device

When a UPnP device is plugged into the network, it has to send the SSDP ANNOUNCE message containing the name of the network service it provides, the network location and protocol to be used to communicate with it. The SSDP server of the bridge is listening to the well-known multicast port number. Thus for each incoming ANNOUNCE message, the bridge device will parse the message according to the manner described below.

The process for integrating a UPnP device into the network is illustrated by the flowchart of figure 5.

For each detected UPnP device/service, the bridge device installs a DCM/FCM to control this UPnP device/service as explained in the previous sections. The HAVi sub-network has the knowledge of these new elements. Any application in the HAVi sub-network can then control a UPnP target. In our example (cf Figure 3) the A device provides the user on its display with the list of devices in its home network represented by icons. To realize that, the user application of A (an HTML browser) has previously queried the HAVi registry to obtain all the identifiers of DCMs which offer an HTTP API.

The user would like to control the E device using HTML and selects its associated icon. The user application in A establishes the HTTP connection with the associated DCM. The User application then sends the HTTP request to the DCM calling the function DCM::HttpRequest. The DCM then establishes the TCP connection according to HTTP between the bridge device (C) and the UPnP target (E) and forwards the HTTP request. To establish the connection, the DCM will use the URL associated with the HTTP server service of the UPnP target previously registered as configuration data.

Once the HTTP command (the HTTP_GET command for instance) is received by the UPnP target, it sends back the response (an HTML page) to the DCM which will forward this response to the source of the request.

The modelization of a HAVi target seen from the UPnP sub-network will now be described. The process for integrating a HAVi device into the network is illustrated by the flowchart of figure 6.

The UPnP model is based on the TCP/UDP/IP protocols. Consequently, a UPnP device is a network entity which can be reached through its IP (Internet Protocol) address. A service is an entity within the application layer (over TCP or UDP) which can be reached through a port number. The port number identifies the connection path between two UPnP entities (an application and a service for example). To represent a HAVi device or a HAVi functional component within the bridge we basically have two solutions:

- The first solution is based on port numbers and is used for the rest of the description, as explained below.

- The second solution is based on multiple IP addresses: local IP addresses are assigned to each device or functional component within the HAVi network. This assignment can be made to be consistent with the SSDP mechanism. The invoked HAVi component (either a DCM or an FCM) can, on instantiation, request a unique IP address assignment from the bridge just as any other UPnP device entering the UPnP network.

The notion of a 'device' in UPnP is used only to reach the network entity and to obtain data describing the location where the service resides. Consequently there is no need to represent a HAVi device as a UPnP device. What is needed is to represent some HAVi services as the HAVi functional component APIs (FCM) and the HTTP API

(offered by the DCM). According to the present embodiment, the so-called port number solution consists in representing a HAVi service by a proxy UPnP service (like a HTTP proxy server) where each proxy is associated with a port number (either TCP or UDP) attributed by the bridge.

5

UPnP requires that for any new services the SSDP client of the hosted device announce the service list it owns. The bridge uses the HAVi REGISTRY to be able to query or detect the new HAVi targets as FCM or DCM according to the appropriate criteria, which is, according to the present embodiment, the GUIREQ value.

10 Consequently, once the bridge detects a HAVi target (either a DCM or a FCM) for which a proxy UPnP service can be activated, it generates the SSDP multicast ANNOUNCE message to the UPnP subnetwork. The following ANNOUNCE message is used to announce a HTTP server service represented by a HTTP proxy within the bridge device:

15

ANNOUNCE server:HTTP HTTP/1.1 Location: httpu://bridge.com:123

The URL "server:HTTP" is the type of the service. The location field contains the URL to reach the service. It is composed of the address of the bridge device and a
20 port number chosen by the bridge device and dedicated to the HTTP proxy associated with the HAVi target.

Each time a new proxy has to be activated, the bridge device chooses a new (private) port number related to the transport protocol used to reach that service (either UDP or TCP).

25 The entity body of the ANNOUNCE message will contain the following fields which help to identify the device hosting the service:

- Device manufacturer name
- Device name
- 30 - A URL to obtain an icon representing the device

It is possible also that the SSDP server of the bridge device receives an OPTIONS message from any UPnP device. The bridge will have to respond according to the description presented earlier.

35

Consequently the proxy service shall maintain a set of configuration data to be able to identify and join its associated HAVi target (either a DCM or a FCM). This configuration data shall comprise the Software Element Identifier (SEID) of the HAVi target.

Once a proxy is installed and declared to the UPnP sub-network, any application in the UPnP sub-network can then control these HAVi targets. In our example (cf Figure 3) the device D provides to the user the list of devices in its home network represented by icons. To realize that, the user application of device D (a HTML browser) has previously queried the UPnP sub-network through the SSDP OPTIONS method to get the description of all devices which implement a HTTP server service.

The user would like to control the device B using HTML and selects its associated icon. The user application in device D establishes the TCP connection with the HTTP proxy –associated with the HAVi target device B - embedded in the bridge device. The HTTP proxy then establishes a HTTP connection with the DCM representing the HAVi target calling the DCM::HttpOpen method as described earlier. The User application then sends the HTTP request to the HTTP proxy through the TCP connection. The HTTP proxy then forwards the HTTP request to the HAVi target device B (in fact its DCM) by calling the function DCM::HttpRequest. The DCM then sends back the HTTP response (its HTML home page for instance) by calling the HTTP proxy (method <client>::HttpResponse) acting as the HTTP client within the HAVi sub-network. The HTTP proxy will then forward the response to the user application located in device D.

Although the preferred embodiment concerns the interoperability of a UPnP sub-network and of a HAVi sub-network, the invention is not limited to these two network types and may also be applied to other types of networks.

Claims

1. Method for bridging a HAVi sub-network and a UPnP sub-network
5 comprising the steps of:

- discovering UPnP devices and/or services on the UPnP sub-network;
- declaration, by a sub-network bridging device, of a discovered UPnP
device as a HAVi Device Control Module and of a discovered UPnP service as a HAVi
10 Functional Component Module on the HAVi sub-network.

2. Method according to claim 1, wherein the step of declaring a Device
Control Module and Functional Component Module comprises the step of registering in
a Registry of the bridging device.
15

3. Method according to claim 1, wherein the discovery step is carried out
using Simple Service Discovery Protocol (SSDP) functions.

4. Method for bridging a HAVi sub-network and a UPnP sub-network
20 comprising the steps of:

- discovering HAVi software elements of the HAVi sub-network
corresponding to a selection criterion;
- representing, in a sub-network bridging device, each of said discovered
25 elements by a UPnP proxy service identified by a port number attributed by said sub-
network bridging device; and
- announcing each of said proxy services on the UPnP sub-network.

5. Method according to claim 3, wherein the discovery step comprises the
30 step of requesting, by said sub-network bridging device, a list of software elements from
its HAVi Registry.

6. Method according to claims 3 or 4, wherein the step of announcing a
proxy service comprises transmission of a bridging device address and proxy service
35 port number.

7. Method according to one of the claims 3 to 6, wherein the bridging
device maintains a set of configuration data for each proxy service, identifying an

associated HAVi software element, said data comprising the software element's identifier.

8. Method according to one of the preceding claims, wherein the selection
5 criterion is HTTP capability.

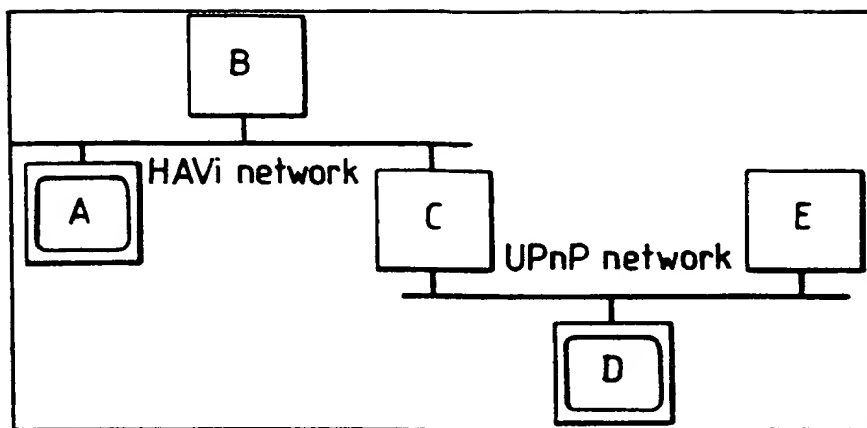


FIG.1

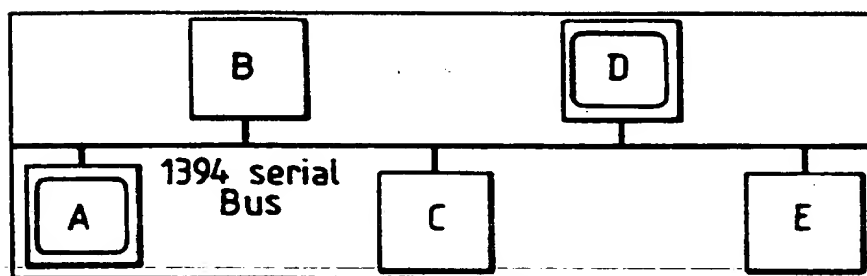


FIG.2

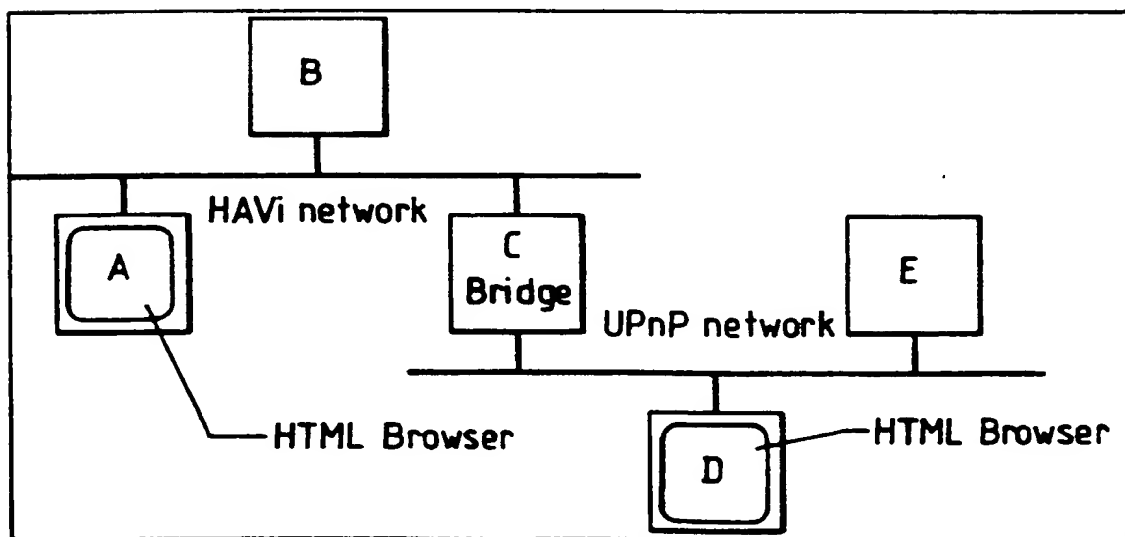


FIG.3

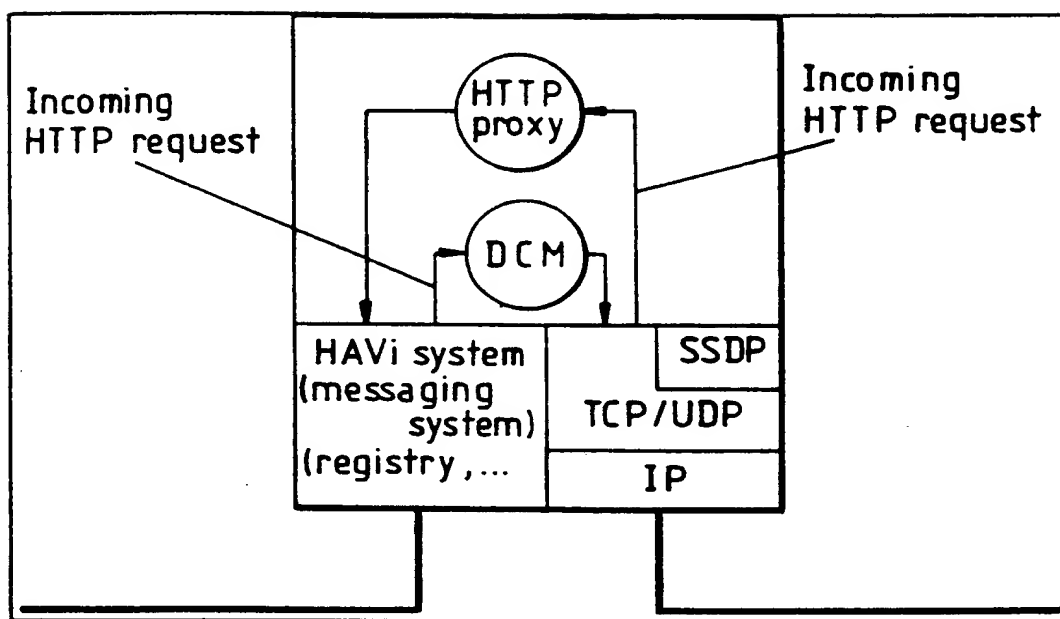
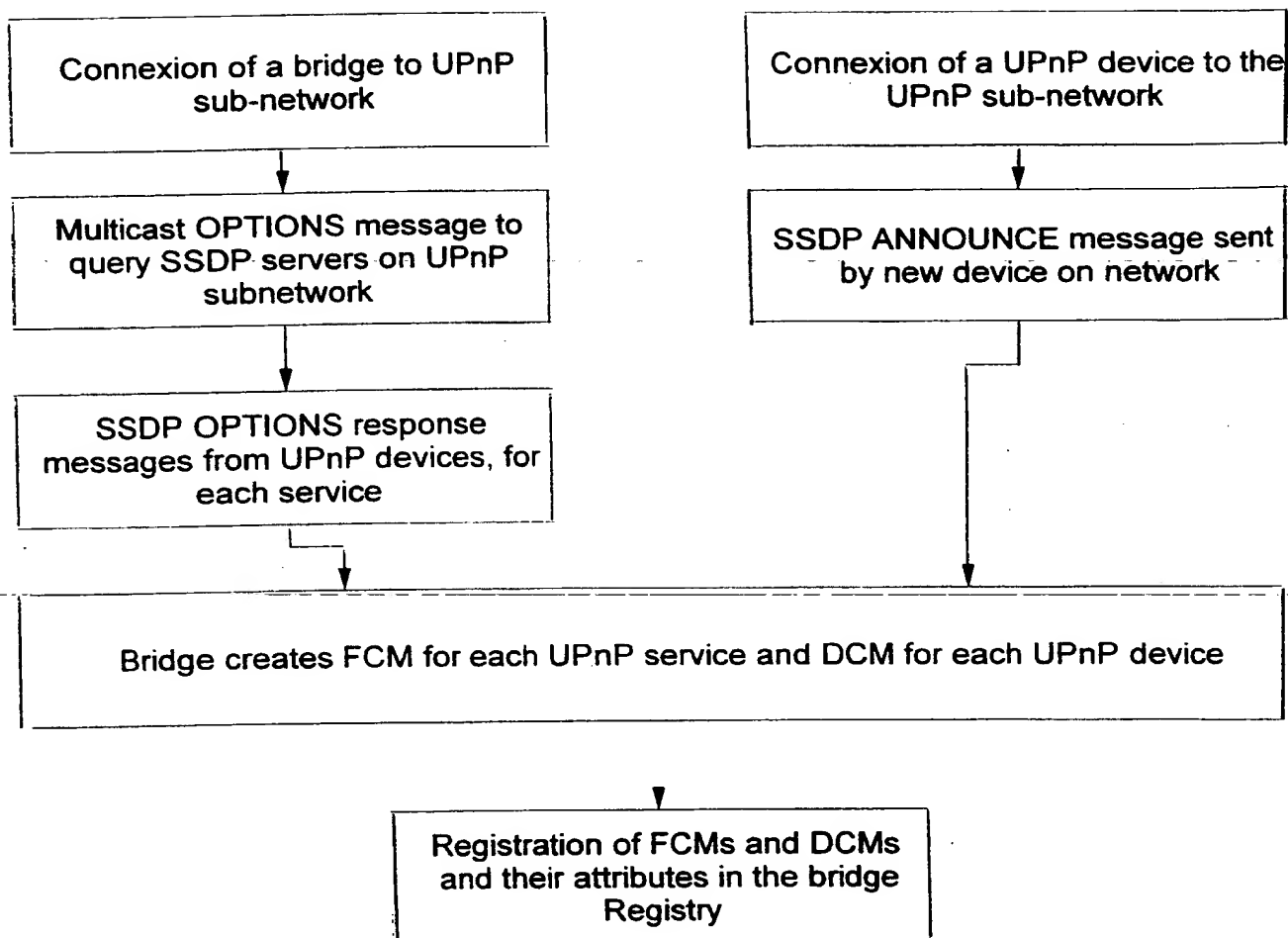
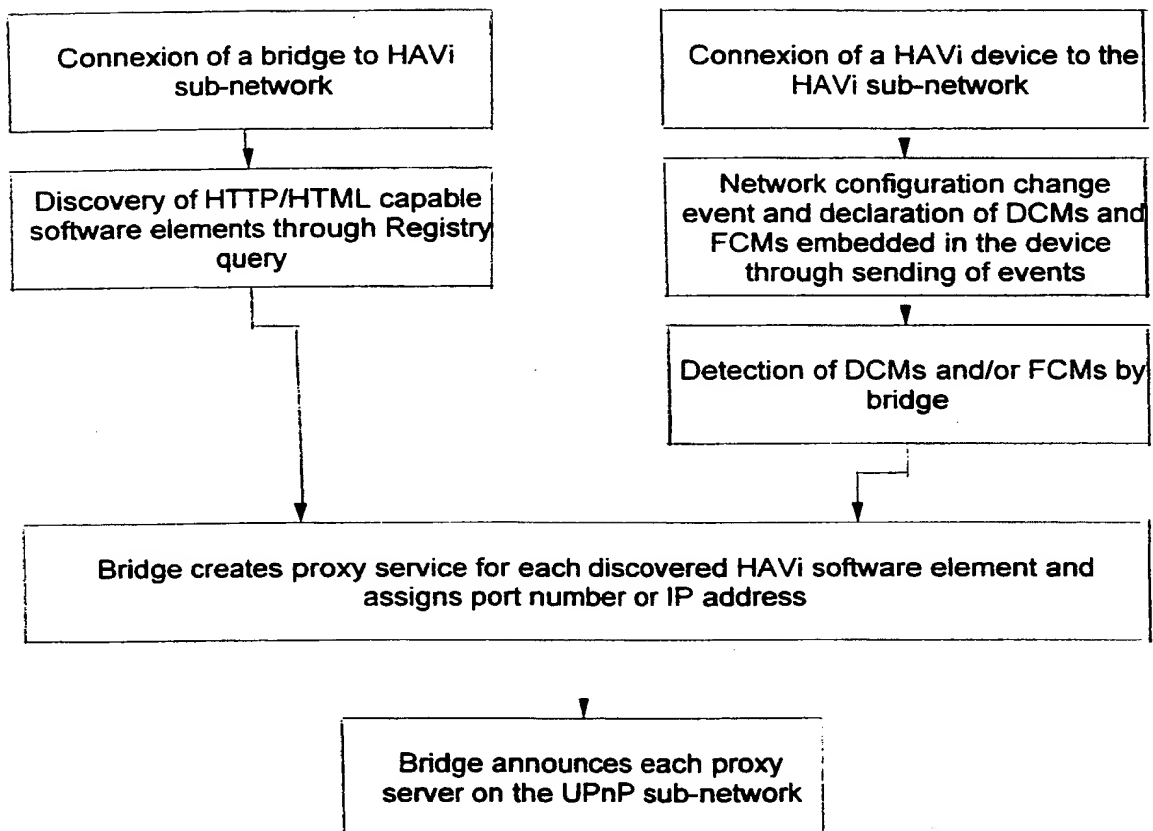


FIG. 4

**Figure 5**

**Figure 6**

INTERNATIONAL SEARCH REPORT

International Application No

PCT/EP 00/05026

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/28

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|------------|--|-----------------------|
| A | EP 0 849 913 A (SONY CORP) 24 June 1998 (1998-06-24) column 1, line 17-27 column 3, line 34-58 column 5, line 47 -column 10, line 52 column 12, line 38 -column 13, line 33 — —/— | 1,4 |

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *G* document member of the same patent family

Date of the actual completion of the international search

5 October 2000

Date of mailing of the international search report

16/10/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5618 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Dupuis, H

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP 00/05026

| C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT | | |
|--|---|-----------------------|
| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
| A | <p>DESBONNET J ET AL: "SYSTEM ARCHITECTURE AND IMPLEMENTATION OF A CEBUS/INTERNET GATEWAY"</p> <p>IEEE TRANSACTIONS ON CONSUMER ELECTRONICS,US,IEEE INC. NEW YORK, vol. 43, no. 4, 1 November 1997 (1997-11-01), pages 1057-1062, XP000768558</p> <p>ISSN: 0098-3063</p> <p>page 1057, left-hand column, paragraph 1 - paragraph 4</p> <p>page 1058, left-hand column, paragraph 1</p> <p>page 1060, right-hand column, paragraph 1 - paragraph 2</p> | 1,4 |
| A | <p>CHILD J: "INTELLIGENT HOME TECHNOLOGY LOOKS FOR LEVERAGE FROM RELATED MARKETS"</p> <p>COMPUTER DESIGN,US,PENNWELL PUBL. LITTLETON, MASSACHUSETTS, vol. 36, no. 12, 1 December 1997 (1997-12-01), pages 85-87, XP000754856</p> <p>ISSN: 0010-4566</p> <p>page 87, left-hand column, paragraph 2 -right-hand column, paragraph 1</p> | 1,4 |
| A | <p>SONY PHILIPS HITACHI SHARP MATSUSHITA THOMSON TOSHIBA GRUNDIG: "Specification of the Home Audio/Video Interorperability Architecture"</p> <p>HAVI ARCHITECTURE,XX,XX, 11 May 1998 (1998-05-11), pages 25-29, XP002115566</p> <p>the whole document</p> | 1,4 |

INTERNATIONAL SEARCH REPORT

information on patent family members

International Application No

PCT/EP 00/05026

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---------------------|-------------------------------|--------------------------|
| EP 0849913 A | 24-06-1998 | JP 10178438 A CA 2224748 A | 30-06-1998 18-06-1998 |

This Page Blank (uspto)